Taha Zulfiqar and Ming Yi Chan
July 15th, 2016

A3 Report

Would you expect the program to run faster with more than one process?

It is expected that a program will run faster with more than one process since the workload can be split and several processes can execute at the same time on different processors, thus decreasing overall run time. Even if only one processor was available, a “blocked” process can make way for a “ready” process to be run. In this case – counting up the number of occurrences of specific word lengths – the run time decrease is most evident at N = 1 (Time = 1.44s) vs. N = 2 (Time = 0.78s) in *Figure 1.*

Why does the speedup eventually decrease below 1?

The speedup does not decrease below 1 in our tests; this is mainly due to the lack of resources available on our machine to execute > 220 fork()s.

However, it would be expected that the speedup eventually decreases below 1 as N gets sufficiently large due to the increasing cost of setting up a pipe for each child process, fork()ing N child processes, and reading in from all N pipes.

Did the performance results surprise you? If so, how?

No. Although we expected the run time to grow as N increased, it became obvious from our early tests (such as with N = 302) that our machine simply cannot handle so many active processes at once (as mentioned above.) Since we cannot test for large values of N, we suspected that we would not be able to observe big growths in running time as N increased.

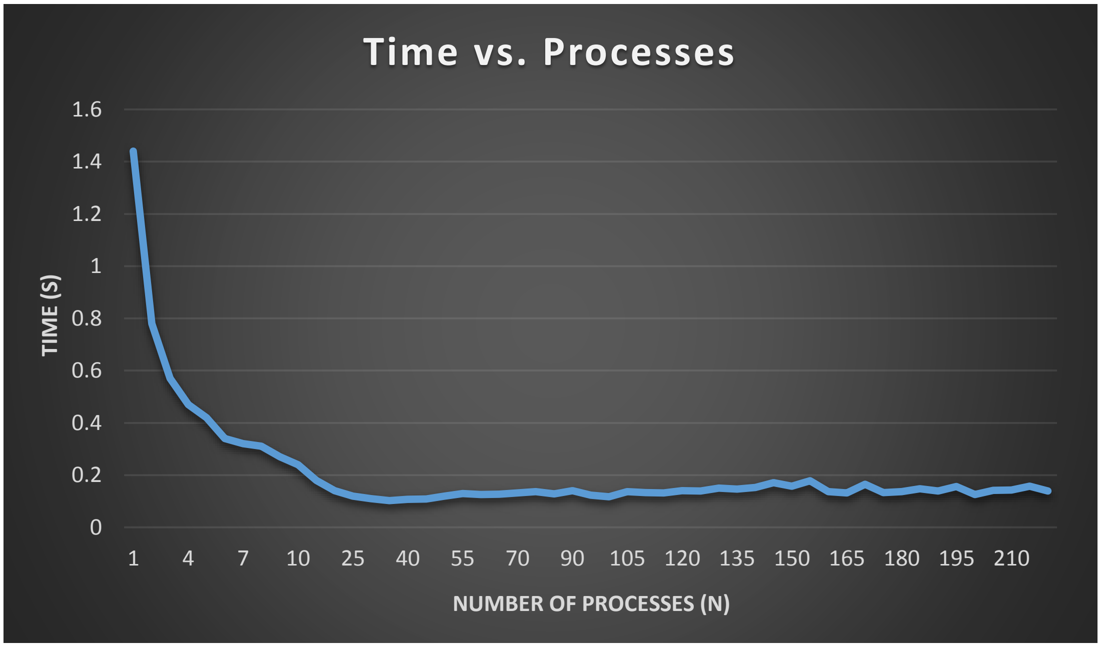

Fig 1. The graph of Time vs Number of Processes